# Red Hat OpenShift Container Storage

## Introduction

This document contains configuring Red Hat OpenShift Container Storage 4.4/4.5 (OCS) on existing Red Hat OpenShift Container Platform 4.5.x (OCP) worker nodes on bare metal or virtual machines. The OpenShift Container Storage operator installation will be using Local Storage operator which will use file system storage of 10GB for monitoring purpose and block storage of 500GB for OSD (Object Storage Daemon) volumes. These OSDs are useful for configuring any application on top of OCS cluster.

The below operators are required to create OCS cluster and deployed through automation fashion.

* Local Storage Operator
* OpenShift Container Storage Operator

## Configuration requirements

The below table shows about all required nodes hardware configuration.

| **Server Role** | **CPU** | **RAM** | **HardDisk1** | **HardDiak2** | **HardDisk3** |
| --- | --- | --- | --- | --- | --- |
| Bootstrap | 4 | 16 | 120 GB | NA | NA |
| HAProxy | 4 | 16 | 150 GB | NA | NA |
| Master | 8 | 64 | 120 GB | NA | NA |
| Worker | 16 | 64 | 120 GB | 10 GB | 500 GB/2 TB |

## Pre-requisites

* Red Hat OpenShift Container Platform 4.5 cluster console is required with the login credentials.
* Availability of any local storage from any storage (i.e Nimble,3PAR, Local Storage) in OpenShift Container Platform.
* OCS installation on OCP 4.5 cluster requires a minimum of 3 worker nodes but OCS should have exact 3 worker nodes which use two more hard disks with 10GB for mon POD (3 in total using always a PVC) + 500GB (or more than 500GB) volume (a PVC using the default “**thin**” storage class) for the OSD volumes. It also requires 16 CPUs and 64GB RAM for each node and worker node hard disk configuration as shown in above figure.

### Scripts for deploying OCS cluster

**NOTE**

BASE\_DIR - is a base directory path for all automated scripts directories are in place and path is /opt/hpe/solutions/ocp/hpe-solutions-openshift/synergy/scalable

This section provides details on the scripts developed to automate the installation of OCS 4.4 operator on the OCP 4.5 cluster. The scripts used to deploy OCS can be found in the installer VM at *$BASE\_DIR/ocs\_installation*.

* **install\_ocs\_operator.py** - main python script which installs Local Storage operator, OpenShift Container Storage operators, creates file system & block storage and also creates SCs, PVs, PVCs.
* **config.py** - This python script is used to convert user input values into program variables for usage by the install\_ocs\_operator.py script.
* **userinput.json** - The userinput.json file needs to be modified as per user configuration and requirements for installing and scaling OCS cluster.
* **config\_secrets.json** – This encrypted file has OCP cluster login credentials and user needs to provide credentials to this file using 'ansible-vault' command.
* **create\_local\_storage\_operator.yaml** – Creates Local Storage operator's Namespace, installs Local Storage operator.
* **local\_storage\_fs.yaml** – Creates file system storage for monitoring OCS cluster.
* **local\_storage\_block.yaml** – Creates block storage for claiming OSD persistent volumes.
* **ocs\_operator.yaml** – This playbook creates OpenShift Container Storage Namespace, block storage for bounding PVC to Storage Class.
* **storage\_ocs.yaml** – This playbook creates storage classes, PVCs, pods to bring up the OCS cluster.
* **scaling\_ocs\_operator.py -** This python script is used to scale OCS cluster by expanding OCS storage with 3 more worker nodes.

### Installing OpenShift Container Storage on OpenShift Container Platform cluster

1. Login to the installer machine as non-root user and browse to python virtual environment as per DG.
2. Update the *config\_secrets.json* file found at *$BASE\_DIR/ocs\_installation* using 'ansible-vault' command as shown below:

The below command is used to open encrypted file config\_secrets.json

$ ansible-vault edit config\_secrets.json

```

# OPENSHIFT\_USERNAME: <OpenShift Container Platform cluster username>

# OPENSHIFT\_PASSWORD: <OpenShift Container Platform cluster password>

```

3) Update the *userinput.json* file is found at *$BASE\_DIR/ocs\_installation* with the following setup configuration details:

```

# OPENSHIFT\_DOMAIN: "<OpenShift Server sub domain fqdn (api.domain.base\_domain)>",

# OPENSHIFT\_PORT: "<OpenShift Server port number (OpenShift Container Platform runs on port 6443 by default)>",

# LOCAL\_STORAGE\_NAMESPACE: "<Local Storage Operator Namespace (local-storage)>",

# OPENSHIFT\_CONTAINER\_STORAGE\_NAMESPACE: "<OpenShift Container Storage Operator Namespace (openshift-storage)>",

# OPENSHIFT\_CONTAINER\_STORAGE\_LOCAL\_STORAGE\_VERSION: "<OCP\_cluster\_version>",

# OPENSHIFT\_CONTAINER\_STORAGE\_FILESYSTEM\_STORAGE: "<Provide 10GiB worker node's drive for file system storage (Ex: /dev/sdb)>" ,

# OPENSHIFT\_CONTAINER\_STORAGE\_BLOCK\_STORAGE: "<Provide 500Gi worker node's drive for block storage (Ex: /dev/sdc)>",

# OPENSHIFT\_CLIENT\_PATH: "<Provide oc absolute path ending with / OR leave empty in case oc is available under /usr/local/bin>",

# "OPENSHIFT\_CONTAINER\_PLATFORM\_WORKER\_NODES": <Provide OCP worker nodes fqdn list ["sworker1.fqdn", "sworker2.fqdn", "worker3.fqdn"]>,

# "OPENSHIFT\_CONTAINER\_STORAGE\_SCALING\_WORKER\_NODES": <Provide OCS worker nodes fqdn list ["sworker4.fqdn", "sworker5.fqdn", "sworker6.fqdn"]>,

# "OPENSHIFT\_CONTAINER\_STORAGE\_BLOCK\_VOLUME": "<Provide base/scale OCS worker node block storage, should be in Gi/Ti (example - 500Gi or 2Ti)>"

```

5) Execute the following command to deploy OCS cluster.

$ cd $BASE\_DIR/ocs\_installation

$ python -W ignore install\_ocs\_operator.py

The output of the above command as shown below:

$ python -W ignore install\_ocs\_operator.py

Enter key for encrypted variables:

Logging into your OpenShift Cluster

Successfully logged into the OpenShift Cluster

Waiting for 1 minutes to 'Local Storage' operator to be available on OCP web console..!!

'Local Storage' operator is created..!!

Waiting for 2 minutes to OCS operator to be available on OCP web console..!!

'OpenShift Container Storage' operator is created..!!

INFO:

1) Run the below command to list all PODs and PVCs of OCS cluster.

'oc get pod,pvc -n openshift-storage'

2) Wait for 'pod/ocs-operator-xxxx' pod to be up and running.

3) Log into OCP web GUI and check Persistant Stoarge in dashboard.

$

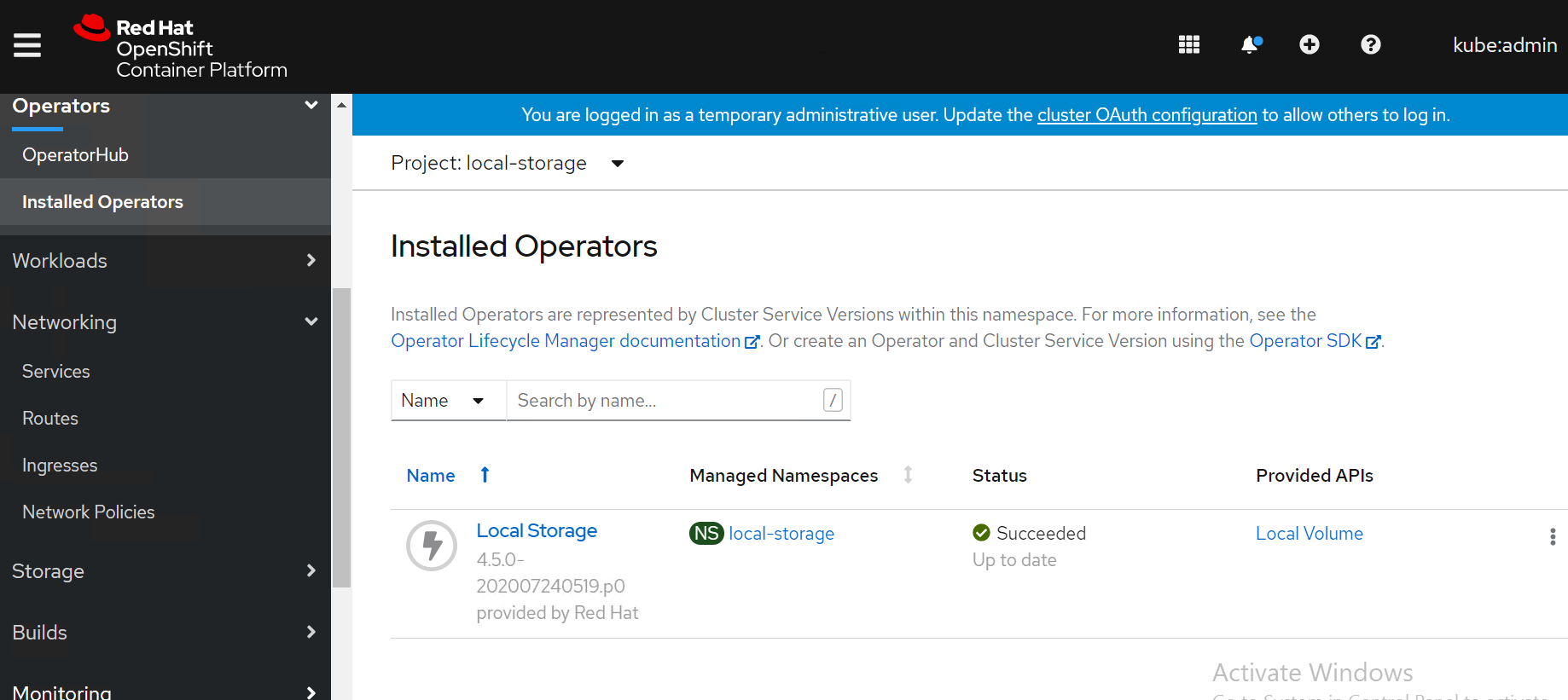
### Validation of the OpenShift Container Storage cluster

The required operators will be created after the execution of the script and they will be reflected in the OpenShift console. This section outlines the steps to verify the operators created through script and are reflected in the GUI:

1) Login to the **OpenShift Container Platform** web console as the user with administrative privileges.

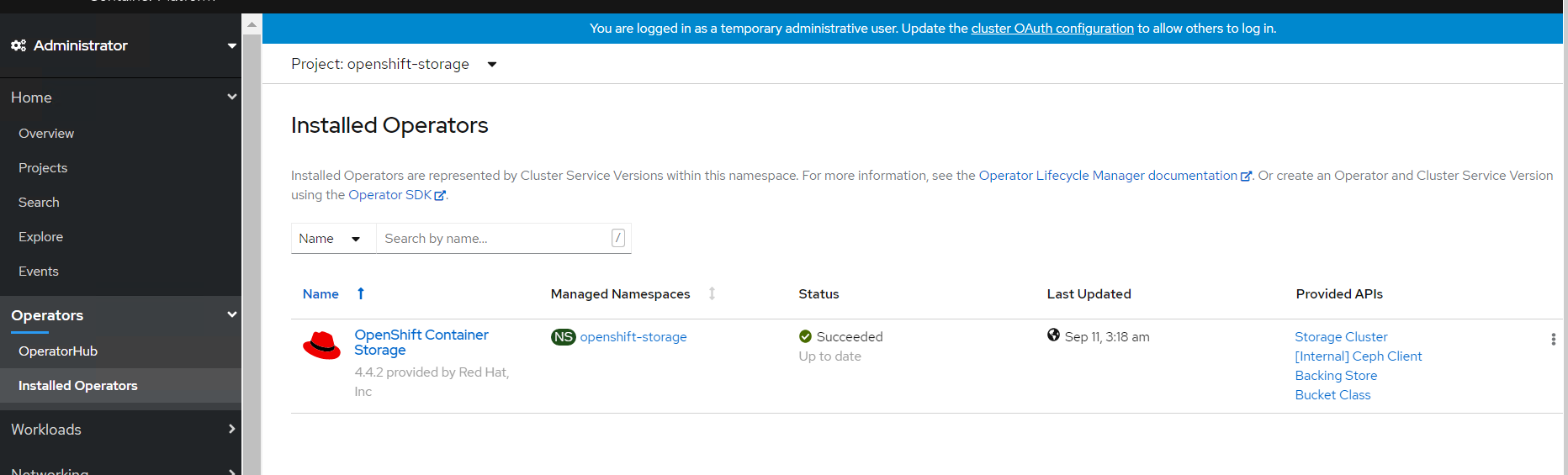
2) Navigate to **Operators -> Project (local-storage)** -> **Installed** **Operators** select your project name.

3) The Local Storage operator will be available in the OpenShift web console as shown in below Figure.



4) Navigate to **Operators** -> **Installed** **Operators** select your project name openshift-storage for OpenShift Container Storage operator.

5) The OpenShift Container Storage operator will be available on the OpenShift Container Platformm web console as shown in below Figure.



6) SCs of OpenShift Container Storage operator on CLI as shown in below.

$ oc get sc

NAME PROVISIONER RECLAIMPOLICY VOLUMEBINDINGMODE ALLOWVOLUMEEXPANSION AGE

local-sc kubernetes.io/no-provisioner Delete WaitForFirstConsumer false 45h

localblock-sc kubernetes.io/no-provisioner Delete WaitForFirstConsumer false 45h

ocs-storagecluster-ceph-rbd openshift-storage.rbd.csi.ceph.com Delete Immediate false 45h

ocs-storagecluster-cephfs openshift-storage.cephfs.csi.ceph.com Delete Immediate false 45h

openshift-storage.noobaa.io openshift-storage.noobaa.io/obc Delete Immediate false 45h

7) PVs of OpenShift Container Storage operator on CLI as shown in below.

$ oc get pv

NAME CAPACITY ACCESS MODES RECLAIM POLICY STATUS CLAIM STORAGECLASS REASON AGE

local-pv-2febb788 500Gi RWO Delete Bound openshift-storage/ocs-deviceset-1-0-lhtvh localblock-sc 45h

local-pv-5f19c0e5 10Gi RWO Delete Bound openshift-storage/rook-ceph-mon-b local-sc 45h

local-pv-b73b1cd5 500Gi RWO Delete Bound openshift-storage/ocs-deviceset-2-0-jmmck localblock-sc 45h

local-pv-b8ba8c38 10Gi RWO Delete Bound openshift-storage/rook-ceph-mon-a local-sc 45h

local-pv-c3a372f6 10Gi RWO Delete Bound openshift-storage/rook-ceph-mon-c local-sc 45h

local-pv-e5e3d596 500Gi RWO Delete Bound openshift-storage/ocs-deviceset-0-0-5jxg7 localblock-sc 45h

pvc-8f3e3d8b-6be7-4ba8-8968-69cbc866c89f 50Gi RWO Delete Bound openshift-storage/db-noobaa-db-0 ocs-storagecluster-ceph-rbd 45h

$

8) PVCs of OpenShift Container Storage operator on CLI as shown in below.

$ oc get pvc

NAME STATUS VOLUME CAPACITY ACCESS MODES STORAGECLASS AGE

db-noobaa-db-0 Bound pvc-8f3e3d8b-6be7-4ba8-8968-69cbc866c89f 50Gi RWO ocs-storagecluster-ceph-rbd 45h

ocs-deviceset-0-0-5jxg7 Bound local-pv-e5e3d596 500Gi RWO localblock-sc 45h

ocs-deviceset-1-0-lhtvh Bound local-pv-2febb788 500Gi RWO localblock-sc 45h

ocs-deviceset-2-0-jmmck Bound local-pv-b73b1cd5 500Gi RWO localblock-sc 45h

rook-ceph-mon-a Bound local-pv-b8ba8c38 10Gi RWO local-sc 45h

rook-ceph-mon-b Bound local-pv-5f19c0e5 10Gi RWO local-sc 45h

rook-ceph-mon-c Bound local-pv-c3a372f6 10Gi RWO local-sc 45h

$

9) PODs of OpenShift Container Storage operator on CLI as shown in below.

$ oc get pod -n openshift-storage

NAME READY STATUS RESTARTS AGE

csi-cephfsplugin-6xpsk 3/3 Running 0 45h

csi-cephfsplugin-7khm6 3/3 Running 0 17m

csi-cephfsplugin-bb48n 3/3 Running 0 45h

csi-cephfsplugin-cfzx6 3/3 Running 0 15m

csi-cephfsplugin-provisioner-79587c64f9-2dpm6 5/5 Running 0 45h

csi-cephfsplugin-provisioner-79587c64f9-hf46x 5/5 Running 0 45h

csi-cephfsplugin-w6p6v 3/3 Running 0 45h

csi-rbdplugin-2z686 3/3 Running 0 45h

csi-rbdplugin-6tv5m 3/3 Running 0 45h

csi-rbdplugin-jgf5z 3/3 Running 0 17m

csi-rbdplugin-provisioner-5f495c4566-76rqm 5/5 Running 0 45h

csi-rbdplugin-provisioner-5f495c4566-pzvww 5/5 Running 0 45h

csi-rbdplugin-v7lfx 3/3 Running 0 45h

csi-rbdplugin-ztdjs 3/3 Running 0 15m

noobaa-core-0 1/1 Running 0 45h

noobaa-db-0 1/1 Running 0 45h

noobaa-endpoint-6458fc874f-vpznd 1/1 Running 0 45h

noobaa-operator-7f4495fc6-lmk9k 1/1 Running 0 45h

ocs-operator-5d664769f-59v8j 1/1 Running 0 45h

rook-ceph-crashcollector-sworker1.socp.twentynet.local-84dddddb 1/1 Running 0 45h

rook-ceph-crashcollector-sworker2.socp.twentynet.local-8b5qzzsz 1/1 Running 0 45h

rook-ceph-crashcollector-sworker3.socp.twentynet.local-699n9kzp 1/1 Running 0 45h

rook-ceph-drain-canary-sworker1.socp.twentynet.local-85bffzm66m 1/1 Running 0 45h

rook-ceph-drain-canary-sworker2.socp.twentynet.local-66bcfjjfkr 1/1 Running 0 45h

rook-ceph-drain-canary-sworker3.socp.twentynet.local-5f6b57c9nt 1/1 Running 0 45h

rook-ceph-mds-ocs-storagecluster-cephfilesystem-a-67fbb67dkb4kz 1/1 Running 0 45h

rook-ceph-mds-ocs-storagecluster-cephfilesystem-b-db66df7dtqkmx 1/1 Running 0 45h

rook-ceph-mgr-a-6f5f7b58dc-fjvjc 1/1 Running 0 45h

rook-ceph-mon-a-76cc49c944-5pcgf 1/1 Running 0 45h

rook-ceph-mon-b-b9449cdd7-s6mct 1/1 Running 0 45h

rook-ceph-mon-c-59d854cd8-gn6sd 1/1 Running 0 45h

rook-ceph-operator-775cd6cd66-sdfph 1/1 Running 0 45h

rook-ceph-osd-0-7644557bfb-9l7ns 1/1 Running 0 45h

rook-ceph-osd-1-7694c74948-lc9sf 1/1 Running 0 45h

rook-ceph-osd-2-794547558-wjcpz 1/1 Running 0 45h

rook-ceph-osd-prepare-ocs-deviceset-0-0-5jxg7-t89zh 0/1 Completed 0 45h

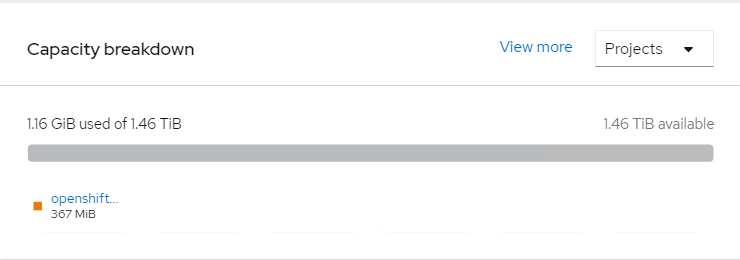
rook-ceph-osd-prepare-ocs-deviceset-1-0-lhtvh-f2znl 0/1 Completed 0 45h

rook-ceph-osd-prepare-ocs-deviceset-2-0-jmmck-wsrb2 0/1 Completed 0 45h

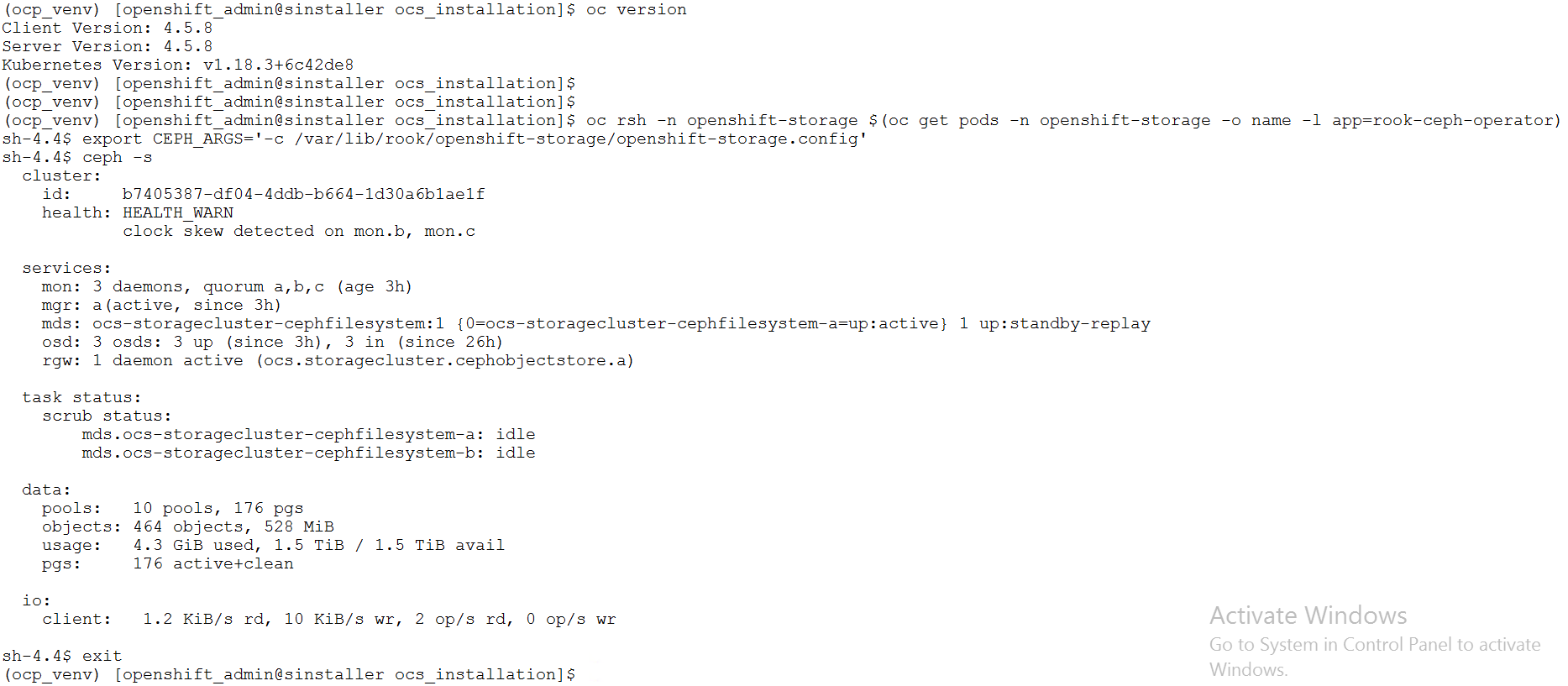
rook-ceph-rgw-ocs-storagecluster-cephobjectstore-a-67b7865qx276 1/1 Running 0 45h

$

10) Storage capacity of OCS cluster with 3 worker nodes (3x500Gi) on OCP web cluster as shown below figure.



11) Storage capacity of OCS cluster with 3 worker nodes (3x500Gi) on OCP CLI as shown below figure.



## Validating OCS with deploying WordPress application

This section covers the steps to validate the OpenShift Container Storage deployment (OCS) by deploying 2-tier application along with MySQL database.

### Prerequisites

1. OCP 4.5 cluster must be installed.
2. OCS to claim persistent volume (PV).

### Deploying WordPress application

**NOTE**

BASE\_DIR - is a base directory path for all automated scripts directories are in place and path is /opt/hpe/solutions/ocp/hpe-solutions-openshift/synergy/scalable

1) Login to the installer machine as a non-root user.

2) From within the repository, navigate to the WordPress script folder

$ cd $BASE\_DIR/ocs\_installation/wordpress

3) Run below script to deploy Wordpress application along with MySQL

$ ./deploy\_wordpress.sh

The deploy\_wordpress.sh scripts does the following activities.

* Creates project
* Sets default storage class
* Deploys Wordpress and MySQL app
* Create routes

5) Below is the output of the scripts

$ ./deploy\_wordpress.sh

Already on project "wordpress" on server "https://api.socp.twentynet.local:6443".

You can add applications to this project with the 'new-app' command. For example, try:

oc new-app ruby~https://github.com/sclorg/ruby-ex.git

to build a new example application in Ruby. Or use kubectl to deploy a simple Kubernetes application:

kubectl create deployment hello-node --image=gcr.io/hello-minikube-zero-install/hello-node

Already on project "wordpress" on server "https://api.socp.twentynet.local:6443".

clusterrole.rbac.authorization.k8s.io/system:openshift:scc:anyuid added: "default"error: --overwrite is false but found the following declared annotation(s): 'storageclass.kubernetes.io/is-default-class' already has a value (true)

service/wordpress-http created

service/wordpress-mysql created

persistentvolumeclaim/mysql-pv-claim created

persistentvolumeclaim/wp-pv-claim created

secret/mysql-pass created

deployment.apps/wordpress-mysql created

deployment.apps/wordpress created

route.route.openshift.io/wordpress-http created

URL to access application

wordpress-http-wordpress.apps.socp.twentynet.local

$

### Verifying the WordPress deployment

1) Execute the following command to verify the persistent volume associated with WordPress application and MySQL database.

$ oc get pods,pvc,route

NAME READY STATUS RESTARTS AGE

pod/wordpress-6f69797b8f-hqpss 1/1 Running 0 5m52s

pod/wordpress-mysql-8f4b599b5-cd2s2 1/1 Running 0 5m52s

NAME STATUS VOLUME CAPACITY ACCESS MODES STORAGECLASS AGE

persistentvolumeclaim/mysql-pv-claim Bound pvc-ccf2a578-9ba3-4577-8115-7c80ac200a9c 5Gi RWO ocs-storagecluster-ceph-rbd 5m50s

persistentvolumeclaim/wp-pv-claim Bound pvc-3acec0a0-943d-4138-bda9-5b57f8c35c5d 5Gi RWO ocs-storagecluster-ceph-rbd 5m50s

NAME HOST/PORT PATH SERVICES PORT TERMINATION WILDCARD

route.route.openshift.io/wordpress-http wordpress-http-wordpress.apps.socp.twentynet.local wordpress-http 80-tcp None

$

2) Access the route url in browser to access the WordPress application as shown below.

